EXPRESS MAIL LABEL NO.: EJ922477644US DATE OF DEPOSIT: 2/11/2000

I hereby certify that this paper and fee are being deposited with the United States Postal Service Express Mail Post Office to Addresee service under 37 CFR §1.10 on the date indicated above and is addressed to the Assistant Commissioner of Patents, Washington, D.C. 20231.

Jennifer Dianne Lane
NAME OF PERSON MAILING PAPER AND FEE

Jennifer Dianne Lane
SIGNATURE OF PERSON MAILING PAPER AND FEE

INVENTORS: K. Attwood, L. Overby, J. Sun

Technique of Defending Against Network Connection Flooding Attacks

Technical Field

The invention relates generally to the field of networking and specifically to defending against attacks by malicious users attempting to disable a server by flooding the server with network traffic.

Background of the Invention

Flooding attacks have recently been used with increasing frequency to target and disable servers on the Internet. A flooding attack occurs when a user sends a large number of requests to a server in a relatively short period of time with an intent to overload and thereby disable the server. A flood of packets from a malicious user can overload a server in the same way that a flood of packets from a misconfigured system can overload a server. But the end result is the same; the server becomes overloaded in trying to service the requests. This prevents legitimate requests from being timely served and often disables a server or causes it to crash. A number of flooding attacks have been reported in the news recently on well known web targets. Flooding attacks are very difficult for traditional intrusion detection systems to prevent due to the difficulty of determining whether traffic is legitimate or not.

Summary of the Invention

The invention recognizes that the consequences of intentional flooding attacks and unintentional overload situations resulting from a burst of connection requests can be mitigated by dropping the traditional notion of attempting to distinguish between legitimate and illegitimate traffic. In the invention, all traffic is subject to a policy that attempts to guarantee that legitimate work will be performed and a server will not crash in flooding situations, irrespective of whether the flooding is caused by legitimate or illegitimate traffic. The invention helps to prevent a server from crashing due to overload and it prevents one or more attackers from consuming all server resources.

In response to a request from a host for a connection to a port number on a server, the number of connections to the port that are assigned to the host are determined. If this number exceeds a first threshold, the request for a connection is denied. In the preferred embodiment, it is possible to override a decision to deny a connection request if a quality of service parameter pertaining to the requesting host permits such an override. However, in the preferred embodiment, if the number of available connections to the port is less than a second threshold, the connection request is denied in any event. The denial of connections to a given host mitigate the effects of intentional or unintentional bursts of connection requests. The overriding of a decision to deny a given request based on a quality of service parameter specific to a requesting host helps in meeting service guarantees that may have been made to a specific host. However, even in the presence of overriding quality of service parameters, the denial of a connection when the number of available port connections becomes prohibitively small helps to prevent the complete disablement of a server.

In the preferred embodiment, the owner of a server specifies for each port number that is subject to flooding checks a maximum number of connections (M) allowed at any time to the port and a controlling percentage (P) of unassigned (available) connections remaining for the port. The invention keeps track of the number of assigned (unavailable) connections to a port and it calculates the number of available port connections by subtracting the number of unavailable connections from the maximum number of connections. The percentage P is used to establish the first threshold to trigger the initial decision to deny a connection request. Specifically, the initial denial is triggered if the existing number of connections assigned to the requesting host is equal to or greater than the threshold percentage of the available connections.

The maximum number of connections and the thresholds will be difficult for most owners to configure. Therefore, a "statistics" mode is provided that measures normal traffic loads of different servers and suggests appropriate maximums and thresholds that will not hamper similar legitimate traffic loads. This statistics mode is not part of the claimed invention and is not described further herein.

A similar technique can be applied to connectionless traffic, such as UDP datagrams. This is the subject matter of patent application number ____.

Brief Description of the Drawing

In the drawing:

Figs. 1 and 2 show an illustrative flowchart of operations executed at a server in response to the receipt of a request for connection to a port to ensure that flooding connection requests do not prevent the completion of other work and do not crash the server.

Detailed Description

The invention requires that an owner of a server using the invention configure the server with certain parameters. By way of example, the preferred embodiment requires that the owner specify for each port number subject to flooding checks a maximum number of connections (M) allowed at any time to the port and a threshold percentage (P) of available connections remaining for the port. The percentage P of available connections for a port establishes a first threshold that triggers the denial of a connection request. As connections are assigned and released, the server maintains the number of connections assigned to each host for each port. The server can therefore dynamically calculate the number of available connections for a port at the time a new request is received from the specified maximum number and the number of connections already assigned to the port.

An entry is made to step 100 in Fig. 1 when a TCP/SYN request for a connection is first received at a network server. A SYN request is the first handshake of a three-flow protocol conventionally required to establish a TCP connection. At step 102, an acknowledgment to the TCP/SYN request is returned by the server to the host requesting a connection. At step 104, the requesting host returns a TCP acknowledgment. This completes the handshake protocol. Step 106 determines the port number to which the request is directed from the requesting host acknowledgment. In TCP, a port number represents a destination within a given host computer to which a connection is requested. Some ports are reserved for standard services. For example, convention specifies that port 21 is used by the File Transport Protocol (FTP). The identity (the IP address) of the requesting host is also determined during the handshake protocol. The port number is used by step 108 to locate a memory control block for the port or to create one if a port control block does not exist. Attached to the port control block are a plurality of host control blocks for hosts that presently have one or more active connections. If the requesting host does not have a host control block, one is created. A host control block contains, among other things, a count of the port connections presently assigned to the host.

At step 110, the server fetches the maximum number of connections M specified for this port number, the controlling percentage P and the number A of active connections. Step 112 calculates the number I of available connections as M - A. Step 114 determines if the number of connections already assigned to the requesting host is equal to or greater than P times I. If so, then the connection request will be denied unless certain other precautions override the denial. On the other hand, if the number of connections already assigned to the requesting host is less than P times I, the connection request is allowed at step 116 and A is incremented by one to update the number of connections active to this port number.

Connection point A in Fig. 2 is entered from step 114 if the number of connections already assigned to the requesting host is equal to or greater than P times I. Step 202 first determines if the port is in a constrained state. A port is in a constrained state if the number of idle connections remaining on the port is equal to or less than some percentage X of the maximum number M of connections allowed to the port. X is 10 percent in the preferred embodiment. If this is true, the connection request is rejected at step 208. However, if the port is not constrained, then a Quality of Service (QOS) specification that pertains specifically to the requesting host can override the decision to reject the connection. In this case, the request might be allowed at step 206, in which case the parameter A is updated by incrementing it by one. In other words, steps 202, 204 and 206 in conjunction implement a policy that rejects a connection request, unless a QOS policy pertaining to the requester overrides the denial. But, if the requested port is in a constrained state, meaning that only a small number of connections remain to the port, the request is denied in any event.

The computer program that has been described can be executed on virtually any type of computer, ranging from personal computers to large mainframes such as IBM's System 390 machines. The only requirement is that the computer is configured with network communication software and is accessible as a server via a network.

Skilled artisans in the fields to which the invention pertains will recognize that numerous variations can be made to the embodiments disclosed herein and still remain within the sprit and scope of the invention.